The Completeness of NAND Gates

In this project, you will implement several combinatorial logic functions using only NAND gates. NAND gates are *complete*; they can be used to build any combinatorial function. This is useful in a practical sense because NAND gates turn out to be easier to implement with transistors than other types of logic gates. Interestingly, JK flip-flops can also be made entirely out of NAND gates as well.

Equipment

- Cadet board
- 7400 Quad NAND DIP (x3)
- wire

Procedure

On the Cadet board, make sure to turn off the power before building your circuit. You will be using the logic switches on the bottom and the LED logic indicators on the right. Make sure these are both set to 5V with the control switches, and that the logic indicators are set to TTL. If you set these incorrectly, the logic high indicators (green LEDs) will not light up.

The logic family used by the components in this project is TTL (Transistor-Transistor Logic). All TTL components have a +5V supply (usually denoted V_{CC}) and a ground connection. The naming scheme for TTL chips is somewhat complicated. The “74” refers to the range of temperatures the device is designed to operate in. The other two or three numbers denote the device's function. There are usually a few letters between the temperature number and the function number, which indicates what “flavor” of TTL logic is used. Other numbers and letters indicate manufacturer, variations, etc. Many TTL components are DIPs (Dual Inline Packages). On most DIPs, there is a dot marking the first pin, and a semicircle marking the top (pin 1 is at the top left corner).

Take 3 7400s and put them in the breadboard, making sure that pin 1 is at the top. On the 7400, ground is pin 7 and Vcc is pin 14. Now connect the NAND gates to implement the logic functions AND, NAND, OR, NOR, XOR, and EQ. An example of how to implement OR and NOR is given. Logic diagrams are shown for all of the functions.

After wiring up the gates, connect the outputs to the LEDs and the inputs to the switches. Verify that your implementation satisfies the truth tables.

When you have your implementation tested and checked off, put everything away in the correct bins and leave the lab better than you found it.

Further Reading

- http://en.wikipedia.org/wiki/Sheffer_stroke
  - This is a fairly in-depth treatment of NAND logic. The article derives a complete logic system using just NAND (a.k.a. Sheffer Stroke). The links go to more digital systems topics.
- http://www.play-hookey.com/digital/jk_nand_flip-flop.html
  - This shows how to make a JK flip-flop with NAND gates.

A	*B*	*A AND B*	*A NAND B*	*A OR B*	*A NOR B*	*A XOR B*	*A EQ B*
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1